

this scheme, the 255th board manufactured within block two would be given the MAC address "OUI:02:00:FF," where OUI is the three-byte hexadecimal value assigned to your company by the IEEE.

### What else?

This month in the Internet Appliance Design section, we've got a very timely set of articles for you. The first, by Greg Wickham, tells us about one company's early experiences with the Java programming language. In this case, the company is Alcatel and the Java code is compiled to native code as part of the development process. In other words, they're using Java without a Java Virtual Machine. Stay tuned folks, I believe ahead-of-time compilation like this has the potential to revolutionize high-end embedded systems development within just a few years.

On the subject of wireless communications, we've got an article about IrDA, by Charles Knutson. This article drills right down into the IrDA protocol stack, showing what you need to implement to support infrared communications with PCs, printers, and other systems. The author focuses on the use of IrDA in resource-constrained embedded systems by showing us what can be left out.

Next month I'll tell you everything you ever wanted to know about the mappings between IP addresses and Ethernet addresses and the corresponding Address Resolution Protocol (ARP). In the meantime, stay connected...

esp

*Michael Barr is the technical editor of Embedded Systems Programming. He holds BS and MS degrees in electrical engineering from the University of Maryland. Prior to joining the magazine, Michael spent half a decade developing embedded software and device drivers. He is also the author of the book Programming Embedded Systems in C and C++ (O'Reilly & Associates). Michael can be reached via e-mail at mbarr@cmp.com.*

### References

1. Frankly, I can never remember which is which: big endian or little endian. So I tend to use the terms "human readable" and "not human readable" on a day-to-day basis, especially when I'm troubleshooting problems alongside other engineers who may be even less versed in the formal terminology.
2. Of course, a convention must also be agreed upon for the application-level data payload being transmitted over the network. However, in this case it is up to the designer(s) of the application to choose the endianness. They'll usually tend to make the choice that minimizes byte reordering.
3. MAC is short for media access control.
4. It should be clear that there is nothing to preclude your storing the hardware address in any nonvolatile memory device, including the ROM or Flash that contains the firmware. This is a decision, however, that should be made with manufacturing concerns, and costs, in mind.
5. I was shocked to learn recently that Cisco owns 60 such sets, for a grand total of over 1 billion addresses!

### Resources

If you're working with TCP/IP, there are two series of books that you should know about. The first is a series of four books by Douglas Comer. In theory, you only need to buy three of these, since there are two versions of Volume 3, but some cross-platform developers may want to get all four. The first volume of the book gives you the big picture of TCP/IP network architecture and the relationships among and between the various protocols. Volume 2 is an implementer's guide, with the most recent edition geared toward ANSI C. The third volume focuses on application-level programming, where you're using the TCP/IP protocols through either the UNIX or Windows sockets API.

Comer, Douglas E. *Internetworking with TCP/IP—Volume 1: Principles, Protocols, and Architecture*, 4th ed.

Englewood Cliffs, NJ: Prentice Hall, 2000.

Comer, Douglas E. and David L. Stevens. *Internetworking with TCP/IP—Volume 2, ANSI C Version: Design, Implementation, and Internals*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1998.

Comer, Douglas E. and David L. Stevens. *Internetworking with TCP/IP—Volume 3, Windows Socket Version: Client-Server Programming and Applications*, 1st ed. Englewood Cliffs, NJ: Prentice Hall, 1997.

Comer, Douglas E. and David L. Stevens. *Internetworking with TCP/IP—Volume 3, BSD Socket Version: Client-Server Programming and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1996.

The second series of books is by W. Richard Stevens (no relation to David L.). The first two volumes of these are my personal favorites, despite the fact that I originally cut my teeth on TCP/IP at a time when only Comer's first edition books were available. This three-volume set follows a similar flow to the Comer books. However, I haven't found as much use for Stevens' Volume 3 as I have for Comer's.

Stevens, W. Richard. *TCP/IP Illustrated—Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.

Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated—Volume 2: The Implementation*. Reading, MA: Addison-Wesley, 1995.

Stevens, W. Richard. *TCP/IP Illustrated—Volume 3: TCP for Transactions, HTTP, NNTP, and the Unix Domain Protocols*. Reading, MA: Addison-Wesley, 1996.

In addition to acquiring at least the first two volumes of one of these series of books, embedded programmers who are new to the world of TCP/IP should also familiarize themselves with the online RFCs. The Internet protocols have always been defined and redefined through written Request for Comments (RFCs). To search the database of current RFCs, try [www.faqs.org/rfcs/](http://www.faqs.org/rfcs/).